

The Effect of Competitor Interaction on Startup's Product Development

Nirnaya Tripathi, Pertti Seppänen, Markku Oivo, Jouni Similä, Kari Liukkunen

M3S Research Unit
University of Oulu
Oulu, Finland

{nirnaya.tripathi, pertti.seppanen, markku.oivo, jouni.simila, kari.liukkunen}@oulu.fi

Abstract— [Context and motivation] Due to lack of resources and teams with low levels of experience, startups face several challenges during their product development, such as product customization, attracting new customers, and mastering the technology uncertainty. To increase their market presence and compensate for their lack of resources, startups need to consider other options such as joint ventures and partnerships. [Question/problem] Some companies that share highly similar resources and businesses can be potential competitors with one another. The effect of interaction with such potential competitors with respect to startups to obtain expertise has not been often reported in the literature to date. [Principal ideas/results] In this study, we simulated two software startups in a controlled experiment to evaluate the effect of interaction with a potential competitor in the effort estimation process. A real startup case was also involved in analyzing the effect. The results of the study show that there is a statistically significant difference in the effectiveness when co-operating with a competitor in the process. Experiment participants also considered the interaction with the potential competitor useful based on the exchange of important information and ideas regarding the product domain. [Contribution] This paper contributes by demonstrating the effect of interaction with a potential competitor in the effort estimation process. In addition, our study encourages further research on startups working in with the competitors in other software engineering knowledge areas.

Keywords— *Startups; Effort estimation; Requirement elicitation; Software process; Competitor*

I. INTRODUCTION

Startups are usually innovative and are considered to be new, inexperienced organizations that lack human and financial resources [1]. With respect to product development in startups, Crowne in [2], discussed four life-cycle stages: startup, stabilization, growth, and evolution. Wang et al. in [3] mentioned the following product development stages: concept, in development, working prototype, functional product with limited users, functional product with high growth, and mature product. During these stages, a startup experiences several challenges, which often leads the company to failure. These challenges include product customization, attracting new customers, managing technological uncertainty, and managing new requirements. [2], [3] Most of the challenges are due to lack of resources and teams with limited experience. To

increase their market presence and compensate for their inadequate resources during product development, startups could consider other options, such as joint ventures and partnerships.

In the context of joint ventures and partnerships, concepts such as that of a “software ecosystem” have been proposed. In a software ecosystem, interested parties such as customers, contractors, companies, and possibly competitors join together to create a relationship on a common platform or in a common market in order to share information and resources to produce better products or services [4]. Collaboration with a Competitor can provide following advantages [5]

1. To obtain new technologies and/or expertise.
2. To minimize the investment in terms of expenses and risks involved in entering new business and markets.

Therefore, partnership with a potential competitor may be a valuable strategy for a startup. Bergen et al. [6] describe a potential competitor as a company that shares high resource similarity but low market commonality. Market commonality refers to the extent to which a given competitor overlaps with the initial company with respect to serving the customer need. Resource similarity refers to the amount of resources the competitor holds with respect to strategic talent in comparison to the initial company.

One example of partnership with a potential competitor is discussed in [7], in which two companies, A and B (both small and medium-sized enterprises (SMEs)), working in the same product domain collaborated in two types of partnership, existing product integration and joint new product development. Company B's main motive for partnership was to enter the new market segment through company A, whereas Company A's primary motive was to use the superior features of Company's B product to improve its own product. The same kind of strategy could be followed by startups as SMEs, since SMEs and startups exhibit some similar characteristics, such as small number of employees and limited resources [1], [8]. Therefore, startups can consider collaboration with a potential competitor to obtain new technologies and/or expertise to improve its existing products and overcome the challenges they face during the product development. However, scientific studies examining the use of potential

competitor's expertise and its effects on the startup's product development are lacking [1] [9].

In this paper, we examine the effect of potential competitor expertise when interacting with the internal team of a startup during the effort estimation process in a simulated environment. We selected the effort estimation process because it is important during product development. Accurate effort estimation may lead to significant cost reduction and improvements in resource planning and assessing risk factors during product development. Therefore, participation of experienced stakeholders during the estimation process is vital. [10], [11] In this case, the potential competitor may prove to be a valuable stakeholder by providing expertise in effort estimation. We propose the following causality research question (RQ) [12] to examine the effect.

RQ: *Does a potential competitor's interaction with a startup's internal team improve the effort estimation process of the startup?*

To answer the causality research question, we used controlled experiment as our research method since they allow the researchers to determine the relationship between variables and cause-effect relationship among them [12], [13]. In the experiment, we simulated two startups in a Master's level course using students as subjects and also used a real startup company. The simulated startups were designed based on Startup characteristics as discussed in [1] and the use of RE practices in a small setting as mentioned in [14], [15]. The reason we simulated two startups in a Master's level course using students as subjects was that students are quite similar to the industry group when evaluating a new technique and their use is common when conducting empirical research in requirements engineering [16], [17]. The two simulated startups were configured in such a way so that they could exhibit as a potential competitor to each other. A real startup case was also involved in comparing and validating the results from the two simulated startups.

The rest of the paper is structured as follows: section 2 provides the background and motivation of our study; in section 3, we discuss the experimental design and execution; in section 4, we discuss the analyses of the data and report the results from the experiment; in section 5, we discuss our research question, compare the results with real-life projects, benefit in collaboration with Competitors, and address the validity of our research; finally, in section 6, we discuss the conclusion of the paper and directions for future work.

II. BACKGROUND AND MOTIVATION

Startups are mostly innovative and exhibit characteristics such as a new and inexperienced organization, inadequate resources, influences from several sources (such as investors, customers), dynamic technologies, and vibrant markets [2]. According to Steve Blank [18], startups can be categorized into four types (based on the market type): 1) *bringing a new product into an existing market*; 2) *bringing a new product into a new market*; 3) *bringing a new product into an existing market and trying to resegment that market as a low-cost entrant*; and 4) *bringing a new product into an existing market and trying to resegment that market as a niche entrant*.

As described in the introduction, Crowne [2] describes product development into four life-cycle stages (startup, stabilization, growth, and evolution), while Wang et al. [3] classify product development into six stages (concept, in development, working prototype, functional product with limited users, functional product with high growth, and mature product). Challenges faced during product development include product customization, mastering technological uncertainty, attracting new customers, and managing new requirements [2], [3]. Most of these challenges can be attributed to scarcity of resources and low levels of team experience in product development. Prior research has also revealed a potential unbalance of competencies between the founder of a startup and the rest of the initial team during the early phases of product development [19].

When a startup is trying to bring a new product into an existing market, the users and the market may be better known along with competitors for the startup [18]. Thus, in order to compensate for a startup's inadequate resources during product development, they could consider joint ventures and partnerships with competitors. In partnership, a startup can collaborate with a potential competitor to improve its own products through integration of the partner's superior features or use of the partner's superior expertise. In return, the startup needs to exhibit reimbursements and benefits of the partnership to the potential competitor. Another benefit also be to reduce the investment risks involved in entering the new market or business.

An important aspect of software development is requirement elicitation. A poor requirement elicitation (RE) procedure and inaccurate effort estimation may lead to complete failure of a software project. During RE, participation of stakeholders with adequate experience and expertise in the application domain is desirable, as the stakeholder can help improve the accuracy of the effort estimation. An accurate estimation of effort during RE is also essential for proper resource allocation and drawing up an appropriate project schedule. [10],[11] However, the requirement engineering process in startups involves only a few elementary activities [1],[14]. To advance the requirement process, some authors [1], [2] have proposed using techniques such as user stories and estimating the effort of each user story.

The most frequent estimation techniques reported in agile software development are expert judgement, planning poker, and use case points. Story points and use case points are often used as size metrics by effort predictors. [20] Improved estimation accuracy during the development effort may lead to significant cost reduction as well as enhanced support in resource planning and assessing risk factors [11].

In this study, we examine whether using the expertise of a potential competitor actually improves a startup's effort estimation accuracy. To do so, we simulated a setting in which the internal team of a startup collaborated with a potential competitor in the effort estimation process. To analyze the effect of this relationship, we designed and conducted a controlled experiment, which is discussed in the following section.

III. RESEARCH DESIGN

The experimental design and execution were carried out according to [13] and reported in the following section based on the guidelines provided by [21].

A. Experiment Planning

1) *Goal definition*: The objects in the study were the startup's internal team and a potential competitor's development team. The purpose was to evaluate the accuracy of the effort estimation in the RE process with respect to the potential competitor's approach. The quality focus was the effectiveness of the two teams. The perspective was that of the researchers. The experiment was performed using M.Sc. students as subjects in a software development project with RE. The study was conducted as a blocked subject-object study, as there is more than one subject and object in the study. [13]

2) *Experimental units*: To conduct the experiment, we used students as experimental subjects, which is common when conducting empirical research in requirements engineering; students are also quite similar to the industry group when evaluating a new technique [16], [17]. In order to simulate a startup environment and conduct the experiment, we planned and executed two software development exercise projects during the autumn of 2015 in the Software Development in Global Environment (SDGE) course [22] at the University of Oulu. The experiment was an obligatory part of the exercise projects. Fifty-nine students registered for the course. Out of fifty-nine students, twenty-six opted for an exam while the other thirty-three elected to pass the course by participating in the exercise projects. During the course, the software development project was introduced to them and participants were allocated to two projects using stratified random sampling [13]. A survey was conducted before the experiment to evaluate participants' experience in software development and requirement engineering.

Three categories were defined based on the participants' work experience in software development and requirement engineering: Experienced (i.e., more than two years of work experience), mid-experienced (i.e., 1–2 years of work experience), and novice (i.e., zero or less than one year of work experience). Based on the survey information, seven participants were placed in the experienced category, 10 participants in the mid-experienced category, and 16 participants in the novice category. To increase the participants' motivation, equal bonus points were awarded to each student, which were added to their SDGE course grades. The two separate software development projects (projects 1 and 2) were carried out in parallel. Each project consisted of a startup, a user team, and a customer team. We included customer and user teams in the RE process for two reasons. Firstly, we are focusing on a scenario in which the startup is trying to bring a new product into an existing market [18]. Secondly, many authors [1], [2] have acknowledged the benefits of including both customers and users in the startup context. Each startup consisted of a product management (PM) team and a development (DEV) team [14].

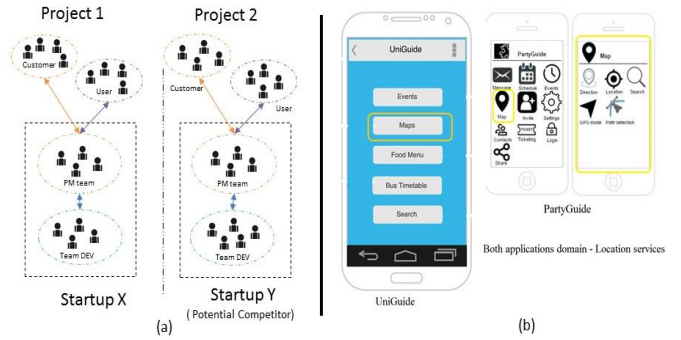


Fig. 1. (a) Two projects and (b) target applications

TABLE I. TEAMS, THEIR TASKS, AND THEIR WORK EXPERIENCE.

Team	Tasks	Experience
Customer	Specifying requirements to the product management team. Collaborating with the user team to prioritize the requirements.	Less than a year
User	Providing requirements from the user perspective. Assessing the prototype.	Less than a year
Product Management	Cooperating with the customer, user, and development teams to gather requirements and creating backlog and final project plan.	More than 2 years
Development	Giving technical requirements for the application. Making the prototype.	Between 1 and 2 years

The outline of the two projects, distribution of teams, and characteristics of the two startups are shown in Figure 1a. A description of the teams' tasks is shown in Table 1. The participants from each experience category were randomly assigned to different teams. Participants from the novice category (i.e., 16 participants) were assigned randomly to two user teams and two customer teams. Each of these teams had four members. The reason for this allocation was that customers and users do not typically have much software development experience, and we were interested to know their perspective in terms of business and the end use of the application. Eight participants from the mid-experienced category and one participant from the experienced category were randomly allocated to two DEV teams.

The DEV team of startup Y had five members while the DEV team of startup X had four members. The objective for this allocation was that the development team in startups typically has a below-average level of experience in software development [1]. Six participants from the experienced category and two participants from the mid-experienced category were randomly allocated to each of the two Team PMs. The purpose for this allocation was that typically, the product is managed by senior managers or the founder [14], who tend to be the most experienced individuals in the startup.

Apart from this, an Expert team (EXP) [23] was created that included a participant from the real software startup case. The participant in that software startup had more than 10 years of requirement engineering and software development experience and had worked in the startup for four years as software developer. The effort estimation of the EXP was used as a baseline for comparison of the estimations of the other teams. The baseline helps to determine whether our treatment is more effective than the control group. In our experiment, we

were not able to evaluate the correctness of the estimations, for instance, by implementing them, so the expert group’s estimation was used as the baseline instead.

3) *Experimental material*: The two software projects were allocated to two startups: X and Y, as shown in Figure 1a. In each project, the startup was given the task of eliciting the requirements of a software application with the user team and the customer team. The projects were designed to include the key attributes of agile RE practices as well as the main characteristics of a software startup. In Table 2, we compare our experiment’s project setting with some agile RE practices of small-scale industrial settings and the main characteristics of a software startup. The project was conducted in four iterations, as described in Table 3. The settings of projects 1 and 2 were identical, so the configurations and resources of Startups X and Y were similar. However, startups X and Y were given different applications for development.

TABLE II. COMPARISON OF RE PRACTICES AND STARTUP CHARACTERISTICS WITH THE EXPERIMENTAL SETTINGS.

RE practices in a small setting [14], [15], [24], [25]	Experiment’s project setting	Startup characteristics [1]	Experiment’s project setting
Iteration	Four 1-week iterations.	Time pressure	The project needed to be completed in four weeks.
Product backlog	Participants created backlog.	Small team	Teams contain 4–5 participants.
User stories	Participants created user stories.	One product	Each project involved one software application.
Sprint retrospective	At end of each week, a steering meeting was held between the customer and PM teams.	Low experience team	Participant in the development team had 1–2 years of experience.

TABLE III. PROJECT OUTLINE.

Event	Project achievement
Pre-project	A training session was held for the participants to provide understanding of the RE, agile practices, and each team’s role in the project as described in Table 1.
Iteration 1	The customer team provided their requirements for the project to the PM team. The PM team collected user stories from the users and the DEV team and put them in the backlog.
Iteration 2	The PM team asked for the prioritization of the user stories in the backlog from the user and the customer. Effort required for user stories was estimated.
Iteration 3	The DEV team started to build a prototype based on the user stories. The PM team discussed the prototype with customers and users to reevaluate the backlog.
Iteration 4	The prototype was redesigned and finalized by the DEV team based on comments from the Stakeholders in the project. All teams wrote a reflection report.
Post-project	Project reflection was carried out through a seminar.

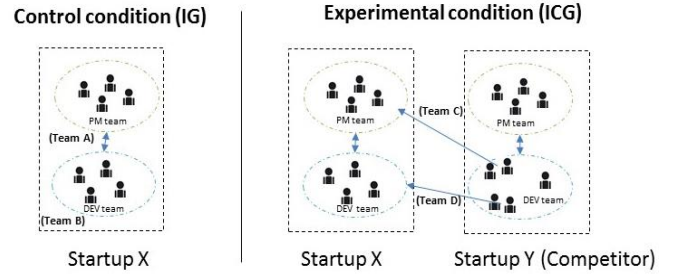


Fig. 2. Control and Experiment groups.

Startup X developed an application called UniGuide for use as a guide on a university campus to provide useful information to students about events and locations. Startup Y developed an application called PartyGuide for use as a guide to various party events happening around the city. Both applications were designed in such a way that they belonged to the same application domain (location services). Because startup X and startup Y were working within the same domain, but on different applications, they can be considered potential competitors [6]. In addition, the similar application domain means that many features of PartyGuide might be of interest to startup X, such as location information, that is, maps (see Figure 1b). Hence, startup X can use the expertise of startup Y in developing their maps feature, if startup X does not have the resources to develop it on their own.

Similarly, startup Y can collaborate with startup X to enter new market segments. Due to these mutual benefits for startups X and Y, they can consider a partnership. In this study, we inspect whether the use of potential competitor expertise actually improves the startup’s effort estimation process for the UniGuide application. To do so, subjects within the PM and DEV teams of startup X were required to develop effort estimates for the user stories they created for UniGuide. We will call this setting the Internal Group (IG).

After that, same subjects within the PM team and DEV team of startup X were required to collaborate with subjects of the DEV team of startup Y to develop effort estimates for the same user stories (see Figure 2). We will call this setting the Internal-Competitor Group (ICG). In addition, the EXP was given the task of developing effort estimates for the user stories created by startup X for UniGuide. The EXP created an effort estimation baseline for comparison to those of the IG and ICG teams to determine which team was closer to the EXP’s estimation.

4) *Experiment Hypotheses*: To answer our research question as stated in Section 1, the following two hypotheses are proposed.

a) *Null hypothesis H1*: The effort estimation (EA) of user stories (US) by the Internal Group (IG) is the same as the effort estimation by the Internal-Competitor Group (ICG).

$H1: \mu (EA)_{US-IG} = \mu (EA)_{US-ICG}$
 (Alternative Hypothesis) $H2: \mu (EA)_{US-IG} \neq \mu (EA)_{US-ICG}$

The objective of H1 is to determine whether there is a significant difference between the effort estimations of IG and ICG. If the alternate hypothesis (H2) holds true, we can move

to determine which group is better. For that, hypotheses H3 and H5 are proposed.

b) Null hypothesis H3: The effort estimation of user stories by the Internal Group (IG) is the same as the effort estimation by the EXP.

$$H3: \mu (EA)_{US-IG} = \mu (EA)_{US-EXP}$$

(Alternative Hypothesis) *H4:* $\mu (EA)_{US-IG} \neq \mu (EA)_{US-EXP}$

H3 is proposed in order to determine whether there is a significant difference between the effort estimations of IG and EXP.

c) Null hypothesis H5: The effort estimation of user stories (US) by the Internal-Competitor Group (ICG) is the same as the effort estimation by the EXP.

$$H5: \mu (EA)_{US-ICG} = \mu (EA)_{US-EXP}$$

(Alternative Hypothesis) *H6:* $\mu (EA)_{US-ICG} \neq \mu (EA)_{US-EXP}$

The objective of H5 is to determine the effectiveness of the ICG in the effort estimation process when compared to EXP. By comparing H3 and H5, we can evaluate which group was more effective in the process of effort estimation.

5) Experiment design: The experiment’s design was based on a block design principle [13], in which the subjects were put into three groups (blocks) based on their experience in order to reduce the effect of their experience on the project. We compared the effort estimations of the ICG with the IG during the effort estimation process. Our experiment can be considered a “one factor, two treatments” design, where the factor was the effort estimation and the treatments were the ICG and the IG. In our research design, each subject identified the effort estimation of the requirements for the software development project during the RE. The dependent variable was measured by analyzing the effort estimation of each user story to determine the effectiveness of the ICG approach.

B. Experiment Operation

1) Tasks performed by the IG: During the second iteration of the project, 32 user stories were created for the UniGuide software by startup X. The subjects of the PM team and the DEV team of startup X were asked to give the effort estimation of each user story in separate meetings. We will call the PM team “Team A” and the DEV team “Team B” to simplify the data analysis (see Figure 2). During both meetings, the effort estimation was carried out via planning poker by giving 1 to 5 story points based on the user story size. During the initial phase of the meeting, a pre-discussion happened between team members in order to acquire sufficient information about the given user story. After that, all the members estimated effort in terms of story points separately and then each member displayed its story point estimation to the others. If the story points differed, then all members, discussed the story until they agreed on a final estimation point for a given user story. Both meetings lasted for two hours.

2) Tasks performed by the ICG: Once the IG performed their task, the subject of the PM team of startup X collaborated with two members of the DEV team of startup Y to generate effort estimates for the user stories. We will call this setting “Team C.” The subject of the DEV team of startup X

collaborated with two other members of the DEV team of startup Y. We will call this setting “Team D” (see Figure 2). The estimation meetings of Team C and Team D occurred separately. During the start of the Team C meeting, the PM team discussed the user stories in the product backlog with two members of the DEV team of startup Y. The PM team leader described the UniGuide software development project to the DEV team of startup Y. Each user story in the product backlog was also clearly described. The estimation process happened just as in IG; however, this time, both teams decided to start with the lowest estimation and then gradually move toward higher estimations. The same process was followed in the Team D meeting. However, members in Team D decided to go through user stories at the top of the backlog first and then gradually moved down. The Team C and Team D meetings each lasted for three hours.

3) Task performed by the EXP: At the end of the second iteration, the 32 user stories collected from startup X were given to the EXP for estimation. The estimation process lasted for ninety minutes. The EXP divided the user stories into three levels: easy, mid, and hard. Easy-level user stories were at a local function level and were easy to implement. Therefore, they were assigned between 1 and 2 points. Mid-level user stories were ones that required lots of testing to achieve the desired featured. Mid-level user stories were assigned between 3 and 4 points. Hard-level user stories required conformance with external systems and more work in comparison to stories at other levels and were therefore assigned 5 points.

IV. ANALYSIS

We present the results in the form of descriptive statistics, cumulative error of effort estimation, and paired sample t-tests for comparison of the effort estimations of the two subject groups to the estimations of the EXP in order to test the hypotheses given in Section 3.1.

A. Descriptive Statistics, Cumulative Error, and Estimation Accuracy

Table 4 presents the sum, mean, and standard deviation differences between the EXP, Teams A and B (IG), and Teams C and D (ICG). In Table 5, the estimations of different teams are presented per user story. The cumulative errors of each team’s estimations compared to the EXP’s estimation are calculated both in absolute points and in percentages on the left-hand side of Table 5. The cumulative errors show that the effort estimation of Teams A and B are nine and eight points larger than that of the EXP, while Team C’s estimates of effort are one point less and Team D’s estimates of effort are eighteen points larger.

TABLE IV. DESCRIPTIVE STATISTICS OF THE TEAMS.

	N	Minimum	Maximum	Sum	Mean	Std. Deviation
Team_Expert	32	1.00	5.00	85.0	2.65	1.40
Team_A	32	1.00	5.00	94.0	2.93	1.18
Team_B	32	1.00	5.00	93.0	2.90	1.20
Team_C	32	1.00	5.00	84.0	2.62	1.40
Team_D	32	1.00	5.00	103.0	3.21	1.12

TABLE V. CUMULATIVE ERROR AND ESTIMATION ACCURACY.

Group	EXP	Cumulative error				Estimation accuracy			
		IG	B	ICG	D	IG	B	ICG	D
US 1	3	4	3	5	4	1	0	2	1
US 2	2	2	3	2	3	0	1	0	1
US 3	3	3	3	3	3	0	0	0	0
US 4	5	5	5	5	5	0	0	0	0
US 5	5	4	3	5	4	1	2	0	1
US 6	2	2	4	2	2	0	2	0	0
US 7	3	3	4	2	4	0	1	1	1
US 8	1	2	1	2	3	1	0	1	2
US 9	2	2	3	3	3	0	1	1	1
US 10	2	3	3	4	2	1	1	2	0
US 11	1	2	2	1	3	1	1	0	2
US 12	1	2	3	1	3	1	2	0	2
US 13	2	2	3	1	4	0	1	1	2
US 14	2	2	1	2	5	0	1	0	3
US 15	3	4	3	3	3	1	0	0	0
US 16	1	2	4	1	2	1	3	0	1
US 17	1	4	4	3	3	3	3	2	2
US 18	3	3	1	2	5	0	2	1	2
US 19	2	1	1	1	1	1	1	1	1
US 20	1	2	3	1	4	1	2	0	3
US 21	1	1	3	1	3	0	2	0	2
US 22	2	3	1	3	2	1	1	1	0
US 23	1	1	3	1	3	0	2	0	2
US 24	3	3	2	2	2	0	1	1	1
US 25	3	3	2	2	2	0	1	1	1
US 26	3	3	2	2	2	0	1	1	1
US 27	3	3	2	2	2	0	1	1	1
US 28	5	5	5	5	5	0	0	0	0
US 29	5	5	5	5	5	0	0	0	0
US 30	5	5	5	4	5	0	0	1	0
US 31	5	4	3	4	3	1	2	1	2
US 32	4	4	3	4	3	0	1	0	1
Sum	85	94	93	84	103	15	35	19	36
Cum error		9	8	-1	18	Error	26	41	22
		11	9%	1%	21%	%	%	%	42%
						50		55	
						58%		65%	

To more closely examine estimation accuracy at the level of a single user story, the absolute errors of each user story as compared to the estimations of the EXP are calculated and summed on the right-hand side of Table 5. The results show that when measuring and improving the accuracy of effort estimation, it is not enough to follow the cumulative efforts because positive/negative errors in estimations may compensate for each other. Instead, the estimation accuracy should be followed by an individual user story. Interestingly, the best cumulative error in our experiment was 1.17% (Team C, ICG), while the error in estimation accuracy was 22% for Team C, ICG, showing that the ICG performed better in our experiment than the IG.

B. Comparison of the ICG and IG with the EXP

To test hypothesis H1, we compared Team A with Team C and Team B with Team D using the paired sample test. As shown in Table 6 on the next page, the mean effort estimation of Team A is 0.31 points more than that of Team C (95% CI [0.062, 0.562]). The significance (2-tailed) value of this pair (Pair 1) is 0.016, which is less than $\alpha = 0.05$. We can thus conclude that there is a statistically significant difference

between the effort estimation of the user story for Team A and for Team C. For pair 2 (Team B and Team D), the mean difference is 0.31 points with a significance (2-tailed) value of 0.169, which is more than $\alpha = 0.05$. This indicates that there is no statistically significant difference between the effort estimations of Team B and Team D. Because the value for pair 1 is less than $\alpha = 0.05$ and their mean difference is significant, we can reject H1.

To test H3, we compared Team A (IG) to the EXP using the paired sample test. Pair 3 in Table 6 shows that the average effort estimation of the EXP is 0.28 points less than that of Team A (95% CI [-0.559, -0.002]). The significance (2-tailed) value in this pair is 0.048, which is less than $\alpha = 0.05$. This demonstrates that there is a statistically significant difference between the effort estimation of the user story for the EXP and for Team A, with a mean difference of 0.28 points. Thus, we can reject H3.

To test H5, we compared Team C (ICG) with the EXP. Pair 4 in Table 6 indicates that the effort estimation of the EXP is 0.03 points more than that of Team C (95% CI [-0.292, 0.354]). The significance (2-tailed) value of this pair is 0.845, which is more than $\alpha = 0.05$. This indicates that there is no statistically significant difference between this pair. Their mean difference is also almost the same. That shows that the effort estimation of Team C is the same as that of the EXP. Further, the value of Cohen's effect size ($d = 0.73$) suggested a moderate to high practical significance for this pair. Thus, H5 holds true.

V. DISCUSSION

In this section, we discuss on our research question, compare the experiment results with real-life projects, benefit in collaboration with competitors, and validity of the study.

A. Answering to the Research Question

To answer our RQ: *Does a potential competitor's interaction with a startup's internal team improve the effort estimation process of the startup?* We analyzed the two experimental groups IG and ICG with EXP. The results indicate that effort estimation by the ICG group was closer to the baseline established by the EXP and was thus better than that of IG. Similarly, we see from Table 5 that both the smallest cumulative error, 1.17%, and the best estimation accuracy, 22%, were obtained by Team C, ICG, showing that the ICG performed better in our experiment than the IG. Based on the above quantitative figures, we can conclude that including potential competitors in effort estimation did improve the effort estimation in our experiment.

Apart from this, the participants' reflection report also confirms the above claim. Participants in the PM team within ICG said that the experience of working in the ICG group was better for them than working in the IG group. Some participants considered the ICG quite useful, as there was an opportunity to ask questions and discuss the user stories in the product backlog. This led to an increase in learning about the product domain requirements and finding appropriate solutions for the problems related to the requirements. As one of the participants in the ICG pointed out:

TABLE VI. PAIRED SAMPLE TEST BETWEEN EXPERT AND THE REST OF THE TEAMS.

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
Pair 1	Team_A - Team_C	.31250	.69270	.12245	.06275	.56225	2.552	31	.016
Pair 2	Team_B - Team_D	-.31250	1.25563	.22197	-.76520	.14020	-1.408	31	.169
Pair 3	Team_Expert - Team_A	-.28125	.77186	.13645	-.55953	-.00297	-2.061	31	.048
Pair 4	Team_Expert - Team_C	.03125	.89747	.15865	-.29232	.35482	.197	31	.845

I was so happy to have this chance to join the development team of startup Y's meeting this week. They received the same tasks as us to estimate efforts for each user story. Though the system we create is different, the main functionalities are very similar. I learned a lot from this meeting and got solutions to what has confused our team for a long time; for example, how to estimate efforts for 'high usability'.

In addition, many of the participants pointed out that the ICG session allowed them to follow the other startup team's working activity and consequently build new ideas. Because members of startup X and Y lack sufficient software development experience, the ICG session helped them to create and share new ideas related to the requirements for the same product domain. For example, one participant in the ICG stated:

All the members of our team actually don't have much software development experience, although members of the development team of startup Y also don't seem to have much experience; therefore, solutions were conducted through sharing ideas and information.

In summary, we found that there is a direct indication that interaction with a potential competitor improves the effort estimation process. This conclusion was further supported by the participant's reflections at the end of the project.

B. Implications of Experiment Results for Real-Life Projects

Taking a closer look at Tables 4 and 5, we discuss the results of our experiment in comparison to effort estimation in real-life projects. A typical real-life project has a defined completion date. Differences between the estimated and actual effort affect this date in two ways: user stories with the actual efforts smaller than the estimated ones loosen the internal schedule, while estimates that are smaller than the actual efforts tighten the internal schedule and may delay the completion date. A looser internal schedule helps to meet the original completion date but may negatively affect the business side. Moreover, the effectiveness of successful product strategies depends on previous market experience and the timing of market entry [26]. An earliest-possible market entry is seen as a key prerequisite for product success, especially for startups. In our experiment, Teams A and B made generous estimations, Team C made a more balanced estimation, and Team D made an over-generous one. In our experiment, we calculated the absolute estimation errors in order to highlight differences that were not visible in the cumulative figures. In real-life projects, absolute estimation errors may vary a great deal depending on factors such as the number of user stories, their complexity, and mutual dependencies.

C. Benefit in collaboration with Competitor

In the literature, such as [2], [3], several challenges were reported during startup product development, such as product customization, attracting new customers, and managing technological uncertainty. These challenges are often due to startup's lack of resources and teams with limited experience. Also, in [5], collaboration with Competitor are highlighted, to obtain new technology, expertise, and avoiding risky investment when targeting new business or market. In our study, we focus on expertise aspect and how it effects on startup's product development. Our study confirms that if a startup collaborates with its potential competitor, it can help them in obtaining the required expertise for their product development. Also, a possible solution for the challenges mentioned above can be looked in the direction of competitor collaboration. For example, in [7], two SMEs that were competitors to each other collaborated to reach their objectives in terms of entering the new market segment and improving the existing product.

D. Threats to Validity

We will discuss the validity of our study based on the four threats to validity described in [13], [16]. With respect to external validity, one threat is that interaction with potential competitors as stakeholders is not common in real-life startups. However, as a startup typically consists of team members who have limited experience in software development, working with potential competitors can help them to share the knowledge and experience required for software development in the product domain. Regarding conclusion validity, one threat could be the low number of startups in our experiment. In this experiment, we divided two startups into four teams and made analyses based on three teams in order to increase the statistical basis to compensate for the low number of startups. Another threat could be the consistency of the data collected during the experiment. However, as the experimental environment is properly defined using the experiment protocol, this threat is decreased.

With respect to construct validity, one threat could be related to hypothesis guessing by the subjects. To minimize this threat, we made sure that the teaching material presented before the experiment did not discuss the context of the experiment. Another threat could be related to evaluation apprehension of the subjects, where they provide high but false estimation points. To overcome this threat, we provided clear instructions before the effort estimation meeting that stated that the subjects need to provide the estimation based on

the credibility of the user stories in the backlog. Regarding internal validity, one threat could be with instrumentation. To overcome this threat, most of the data in the experiment were collected using electronic resources. In addition, the training sessions for the control and treatment groups were kept the same. Another threat could be with regard to the selection of subjects. To overcome this threat, we conducted a mandatory experiment within a course, and students were selected based on stratified random sampling.

VI. CONCLUSION

Interaction with a potential competitor may be valuable for a startup's strategic interest in terms of entering new market segments as well as improving its own existing products by integrating superior features of a competitor's product. In addition, such a competitor may also be a valuable source of requirements during the process of improving existing products. In this paper, we examined this phenomenon in a simulated environment through a controlled experiment. The results of the quantitative data analysis obtained from the experiment show that there is a significant difference in the effectiveness of the potential competitor in the effort estimation process. The descriptive result indicated that subjects with the ICG produced better estimates of effort for user stories when compared to estimates done by the IG subjects. Also, based on the participants' reflections, using a potential competitor in the effort estimation process results in greater information sharing between the startup and the potential competitor, and the participants' perspective regarding requirements for the same software domain did improve. The process also helped less-experienced members to gain valuable knowledge from the competitors' experience regarding product requirements. Based on the results, we believe that interaction with a potential competitor in the RE process will increase exchange of ideas and information sharing related to product domain. There are a number of future perspectives suggested by our study. In this study, we focused on the estimation of the gathered user stories. It would also be interesting to study the quality and quantity of requirements created during the RE. In this research, we included a potential competitor only in RE, but it would be interesting to study how a potential competitor affects other aspects of software engineering areas such as software construction and Testing.

REFERENCES

- [1] N. Paternoster, C. Giardino, M. Unterkalmsteiner, T. Gorschek, and P. Abrahamsson, "Software development in startup companies: A systematic mapping study," *Inf. Softw. Technol.*, vol. 56, no. 10, pp. 1200–1218, 2014.
- [2] M. Crowne, "Why software product startups fail and what to do about it. Evolution of software product development in startup companies," *IEEE Int. Eng. Manag. Conf.*, vol. 1, pp. 338–343, 2002.
- [3] X. Wang, H. Edison, S. S. Bajwa, C. Giardino, and P. Abrahamsson, "Key Challenges in Software Startups Across Life Cycle Stages," in *XP 2016*, Cham: Springer International Publishing, 2016, pp. 169–182.
- [4] S. Jansen, A. Finkelstein, and S. Brinkkemper, "A sense of community: A research agenda for software ecosystems," *31st Int. Conf. Softw. Eng. - Companion Vol.*, pp. 187–190, 2009.
- [5] G. Hamel, Y. L. Doz, and C. K. Prahalad, "Collaborate with your competitors and win," *Harv. Bus. Rev.*, vol. 67, no. 1, pp. 133–139, 1989.
- [6] M. Bergen and M. a. Peteraf, "Competitor Identification and Competitor Analysis: A Broad-Based Managerial Approach," *Manag. Decis. Econ.*, vol. 23, no. 4–5, pp. 157–169, 2002.
- [7] G. Valença, C. Alves, V. Heimann, S. Jansen, and S. Brinkkemper, "Competition and collaboration in requirements engineering: A case study of an emerging software ecosystem," *2014 IEEE 22nd Int. Requir. Eng. Conf. RE 2014 - Proc.*, pp. 384–393, 2014.
- [8] N. Tripathi, E. Annanperä, M. Oivo, and K. Liukkunen, "Exploring Processes in Small Software Companies: A Systematic Review," in *International Conference on Software Process Improvement and Capability Determination*, 2016, pp. 150–165.
- [9] K. Manikas, "Revisiting software ecosystems Research: A longitudinal literature study," *J. Syst. Softw.*, vol. 117, pp. 84–103, 2016.
- [10] C. Pacheco and I. Garcia, "A systematic literature review of stakeholder identification methods in requirements elicitation," *J. Syst. Softw.*, vol. 85, no. 9, pp. 2171–2181, 2012.
- [11] J. M. Verner, W. M. Evanco, and N. Cerpa, "State of the practice: An exploratory analysis of schedule estimation and software project success prediction," *Inf. Softw. Technol.*, vol. 49, no. 2, pp. 181–193, 2007.
- [12] S. Easterbrook, J. Singer, M.-A. Storey, and D. Damian, "Selecting Empirical Methods for Software Engineering Research," in *Guide to Advanced Empirical Software Engineering*, 2008, pp. 285–311.
- [13] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering*. Springer Science & Business Media, 2012.
- [14] J. Melegati and A. Goldman, "Requirements Engineering in Software Startups: a Grounded Theory Approach," in *In Proceedings 22nd ICE/IEEE International Technology Management Conference*, 2016, pp. 448–454.
- [15] A. Jorge, S. Easterbrook, and G. Wilson, "Requirements in the wild: How small companies do it," *Proc. - 15th IEEE Int. Requir. Eng. Conf. RE 2007*, no. 1090, pp. 121–130, 2007.
- [16] M. Daun, A. Salmon, T. Bandyszak, and T. Weyer, "Common Threats and Mitigation Strategies in Requirements Engineering Experiments with Student Participants," in *Requirements Engineering: Foundation for Software Quality*, 2016, pp. 269–285.
- [17] P. Runeson, "Using students as experiment subjects—an analysis on graduate and freshmen student data," in *Proceedings of the 7th International Conference on Empirical Assessment in Software Engineering*, 2003, pp. 95–102.
- [18] S. Blank, *The four steps to the epiphany: successful strategies for products that win*. BookBaby, 2013.
- [19] P. Seppänen, M. Oivo, and K. Liukkunen, "The initial team of a software startup. Narrow-shouldered innovation and broad-shouldered implementation.," in *To be published in 22nd ICE/IEEE International Technology Management Conference*, 2016, pp. 57–65.
- [20] M. Usman, E. Mendes, F. Weidt, and R. Britto, "Effort estimation in Agile Software Development: A systematic literature review," *ACM Int. Conf. Proceeding*, pp. 82–91, 2014.
- [21] A. Jedlitschka, M. Ciolkowski, and D. Pfahl, "Reporting experiments in software engineering," in *Guide to advanced empirical software engineering*. Springer, 2008, pp. 201–228.
- [22] J. Lappalainen, N. Tripathi, and J. Similä, "Teaching a global software development course," *Proc. 38th Int. Conf. Softw. Eng. Companion - ICSE '16*, pp. 440–450, 2016.
- [23] M. Jørgensen, "A review of studies on expert estimation of software development effort," *J. Syst. Softw.*, vol. 70, no. 1–2, pp. 37–60, 2004.
- [24] K. Schwaber and M. Beedle, *Agile Software Development with Scrum*, vol. 18, 2001.
- [25] K. Beck, "Embracing change with extreme programming," *Computer (Long Beach, Calif.)*, vol. 32, no. 10, pp. 70–77, 1999.
- [26] B. L. Bayus and R. Agarwal, "The Role of Pre-Entry Experience, Entry Timing, and Product Technology Strategies in Explaining Firm Survival," *Manage. Sci.*, vol. 53, no. 12, pp. 1887–1902, 2007.